AMENDMENTS TO THE CLAIMS

No claim is amended. A complete listing of the pending claims is provided below.

1. (Previously Presented) A method for predicting the behavior of a workload across a plurality of nodes, the method comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and

e) outputting the prediction.

2. (Original) The method of claim 1 wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

3. (Original) The method of claim 1 wherein the action of identifying potential data conflicts comprise predicting types of data conflicts.

4. (Original) The method of claim 3 in which the types of data conflicts comprises a read-write conflict.

5. (Original) The method of claim 3 in which the types of data conflicts are based upon types of operations needed to resolve the data conflicts.

6. (Original) The method of claim 3 in which the different types of data conflicts have differing levels of expense associated with operations needed for data conflict resolution.

7. (Original) The method of claim 1 in which the potential data conflicts are at the granularity of a data block.

8. (Original) The method of claim 1 in which the potential data conflicts are identified based upon workload division between sessions.

9. (Original) The method of claim 1 further comprising:

f) selecting a number of nodes;

g) dividing the traced execution of the workload across the number of nodes.

10. (Original) The method of claim 9 in which modulo division is used to divide the traced execution of the workload across the number of nodes.

11. (Original) The method of claim 9 in which the number of nodes corresponds to an anticipated number of nodes for a distributed computing system.

12. (Original) The method of claim 9 in which a modulo class represents a node in the number of nodes.

13. (Original) The method of claim 1 in which the potential data conflicts are used to compute costs of migrating the workload to a distributed system.

14. (Previously Presented) A method for distributing a workload across a plurality of nodes, the method comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and

e) outputting the workload distribution scheme.

15. (Previously Presented) The method of claim 14, wherein the forming the workload distribution scheme comprises determining workload distribution in manner which reduces the potential data conflicts.

16. (Original) The method of claim 14, wherein the workload distribution scheme is based upon data accesses.

17. (Original) The method of claim 16 in which the workload is grouped in the workload distribution scheme to maximize intersection of data access on a same group of nodes.

18. (Original) The method of claim 16 in which the workload is grouped in the workload distribution scheme to minimize intersection of data access across different groups of nodes.

19. (Original) The method of claim 14, wherein the workload distribution scheme is based upon access frequencies.

20. (Original) The method of claim 19 in which data objects accessed by the workload are associated with weighting factors.

21. (Original) The method of claim 20 in which not all the data objects are associated with same weighting factors.

22. (Original) The method of claim 20 in which a weighted correlation is performed between the data objects and entities that access the data objects.

23. (Original) The method of claim 22 in which the entities that access the data objects comprises sessions.

24. (Original) The method of claim 22 in which subsets of the entities that access the data objects are grouped together.

25. (Original) The method of claim 24 in which a data structure is employed to represent an affinity between one of the entities that access the data objects and another of the entities.

26. (Original) The method of claim 14 in which the workload comprises data access upon one or more hierarchical objects.

27. (Original) The method of claim 26 in which tracing the execution of the workload comprises tracing identifiers for the one or more hierarchical objects.

28. (Original) The method of claim 14 in which tracing the execution of the workload comprises tracing identifiers associated with entities that access data.

29. (Original) The method of claim 28 in which the entities comprise sessions.

30. (Original) The method of claim 28 in which the workload distribution scheme distributes the workload based upon partitioning of the entities that access data.

31. (Previously Presented) The method of claim 30 in which an association is formed between partitioning of the entities that access data and partitioning of one or more applications within the workload.

32. (Previously Presented) A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for optimizing the distribution of a workload across a plurality of nodes, the process comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, optimizing the distribution of the workload across the plurality of nodes; and

e) outputting the optimized distribution scheme.

33. (Previously Presented) A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for distributing a workload across a plurality of nodes, the process comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and

e) outputting the workload distribution scheme.

34. (Previously Presented) A system for distributing a workload across a plurality of nodes, comprising:

a) means for receiving a workload to be executed;

b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) means for tracing the execution of the workload to identify a potential data conflict;

d) means for, based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and

e) means for outputting the workload distribution scheme.

35. (Previously Presented) A system for optimizing the distribution of a workload across a plurality of nodes, comprising:

a) means for receiving a workload to be executed;

b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) means for tracing the execution of the workload to identify a potential data conflict;

d) means for optimizing the distribution of the workload across the plurality of nodes based on a result of the tracing; and

e) means for outputting the optimized distribution scheme.

36. (Previously Presented) A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for predicting the behavior of a workload across a plurality of nodes, the process comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and

e) outputting the prediction.

37. (Previously Presented) The computer program product of claim 36 wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

38. (Previously Presented) The computer program product of claim 36 wherein the action of identifying potential data conflicts comprises predicting types of data conflicts.

39. (Previously Presented) The computer program product of claim 36 in which the potential data conflicts are identified based upon workload division between sessions.

40. (Previously Presented) A system for predicting the behavior of a workload across a plurality of nodes, comprising:

a) means for receiving a workload to be executed;

b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) means for tracing the execution of the workload to identify a potential data conflict;

d) means for, based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and

e) means for outputting the prediction.

41. (Previously Presented) The system of claim 40 wherein the means for tracing is configured to predict how many data conflicts will occur.

42. (Previously Presented) The system of claim 40 wherein the means for tracing is configured to predict types of data conflicts.

43. (Previously Presented) The system of claim 40 in which the means for tracing is configured to identify the potential data conflicts based upon workload division between sessions.

44. (Previously Presented) The computer program product of claim 33, wherein the forming the workload distribution scheme comprises determining workload distribution in manner which reduces the potential data conflicts.

45. (Previously Presented) The computer program product of claim 33, wherein the workload distribution scheme is based upon data accesses.

46. (Previously Presented) The computer program product of claim 33, wherein the workload distribution scheme is based upon access frequencies.

47. (Previously Presented) The computer program product of claim 33 in which the workload comprises data access upon one or more hierarchical objects.

48. (Previously Presented) The computer program product of claim 33 in which tracing the execution of the workload comprises tracing identifiers associated with entities that access data.

49. (Previously Presented) The system of claim 34, wherein the means for forming the workload distribution scheme comprises means for determining workload distribution in manner which reduces the potential data conflicts.

50. (Previously Presented) The system of claim 34, wherein the workload distribution scheme is based upon data accesses.

51. (Previously Presented) The system of claim 34, wherein the workload distribution scheme is based upon access frequencies.

52. (Previously Presented) The system of claim 34 in which the workload comprises data access upon one or more hierarchical objects.

53. (Previously Presented) The system of claim 34 in which the means for tracing the execution of the workload comprises means for tracing identifiers associated with entities that access data.

54. (Previously Presented) A method for optimizing the distribution of a workload across a plurality of nodes, the method comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, optimizing the distribution of the workload across the plurality of nodes; and

e) outputting the optimized distribution scheme.

55. (Previously Presented) The method of claim 54, wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

56. (Previously Presented) The method of claim 54, wherein the action of identifying potential data conflicts comprise predicting types of data conflicts.

57. (Previously Presented) The method of claim 54 in which the potential data conflicts are at the granularity of a data block.

58. (Previously Presented) The method of claim 54 in which the potential data conflicts are identified based upon workload division between sessions.

59. (Previously Presented) The method of claim 54 in which the potential data conflicts are used to compute costs of migrating the workload to a distributed system.

60. (Previously Presented) The computer program product of claim 32, wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

61. (Previously Presented) The computer program product of claim 32, wherein the action of identifying potential data conflicts comprise predicting types of data conflicts.

62. (Previously Presented) The computer program product of claim 32 in which the potential data conflicts are at the granularity of a data block.

63. (Previously Presented) The computer program product of claim 32 in which the potential data conflicts are identified based upon workload division between sessions.

64. (Previously Presented) The computer program product of claim 32 in which the potential data conflicts are used to compute costs of migrating the workload to a distributed system.

65. (Previously Presented) The system of claim 35, wherein the means for tracing is configured to predict how many data conflicts will occur.

66. (Previously Presented) The system of claim 35, wherein the means for tracing is configured to predict types of data conflicts.